Continual Learning in Deep Networks: an Analysis of the Last Layer

Timothée Lesort Thomas George Irina Rish

Université de Montréal, MILA - Quebec AI Institute

Abstract

We study how different output layer types of a deep neural network learn and forget in continual learning settings. We describe the three factors affecting catastrophic forgetting in the output layer: (1) weights modifications, (2) interferences, and (3) projection drift. Our goal is to provide more insights into how different types of output layers can address (1) and (2). We also propose potential solutions and evaluate them on several benchmarks. We show that the best-performing output layer type depends on the data distribution drifts or the amount of data available. In particular, in some cases where a standard linear layer would fail, it is sufficient to change the parametrization and get significantly better performance while still training with SGD. Our results and analysis shed light on the dynamics of the output layer in continual learning scenarios and help select the best-suited output layer for a given scenario.

Contribution

- We propose an evaluation of a large panel of output layer types in continual scenarios: we review the capacity of each layer to learn continually or from a subset of samples.
- We describe the different sources of performance decrease in continual learning for the output layer: forgetting, interferences, and projection drifts.
- We review and propose different solutions to address catastrophic forgetting in the output layer. In particular, we introduce a simplified weight normalization layer, two masking strategies, and an alternative to Nearest Mean Classifier using median vectors.

Output Layer Types

For z a latent vector the output layer computes the operation

$$o = Az + b$$

It can be rewritten for a single class:

$$\|z\|\|A_i\| \cdot cos(\angle(z, A_i)) + b_i = o_i$$

Where $\angle(\cdot, \cdot)$ is the angle between two vectors and $\|\cdot\|$ denotes here the euclidean norm of a vector.
Removing the bias (*Linear_no_bias* layer):

$$\|z\|\|A_i\| \cdot cos(\angle(z_t, A_i)) = o_i$$

Normalizing output vectors (**Proposed *WeightNorm* layer**):

$$\|z\| \cdot cos(\angle(z_t, A_i)) = o_i$$

Measuring only the angle (*CosLayer*):

$$cos(\angle(z, A_i)) = o_i$$

Other output layer based on similarity of training embeddings and test embeddings can be used such as KNN (K-nearest neighbor), NMC (Nearest Mean Classifier), LDA (linear discriminant analysis). Those types of classifiers can be very efficient in CL.

Visualization

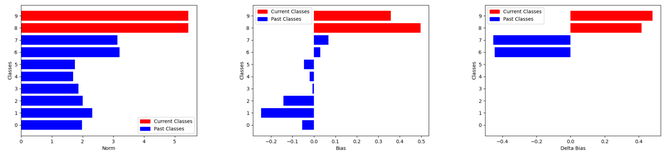

Fig. 1: Illustration of norm (left), bias (middle) and delta bias (right) unbalance at the end of a CIFAR10 continual experiment with a linear layer.

We propose to apply a masking strategy to avoid the modification of past classes while learning new ones. For one observation, the masking either enable the modification of one vector only or to all the vector inside the current batch (group masking).

Fig. 2: Illustration of Forgetting: normal linear layer vs masked layer. Masking avoids the modifying/forgetting of weights of other classes.

Experiments

Incremental Scenarios: CIFAR10 (5 tasks / 2 classes by task), Core50 (10 tasks / 5 classes by task): Evaluate the capacity of learning incremental new classes and distinguishing them from the others.
Lifelong Scenario: Core10Lifelong (8 tasks / 1 env by task / 10 classes by task): The classes stay the same, but the instances change. These settings evaluate the capacity of improving at classifying with new data.
Mixed Scenario: Core10Mix (50 tasks / 1 class by task): In this scenario, a new task is triggered by either a virtual concept drift (new class) or a domain drift (kown class, new data).
Experiments are run on 8 different task orders. (We also experiment with the ability for each layer to learn from few samples but it is not presented in this poster.)

Results: Comparison Gradient Based Methods

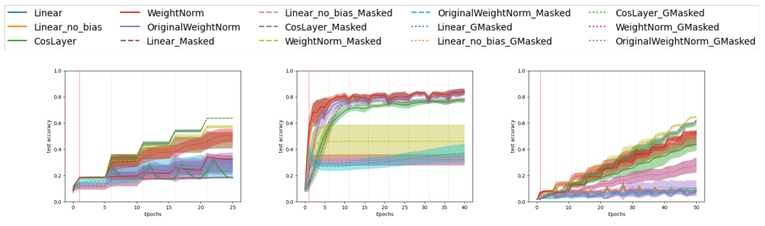

Fig. 3: Experiments on CIFAR10 (left), Core10Lifelong, and Core50 (Right) on 8 different task orders. We plot the test accuracy on the full test set for each epoch. We compare the different parameterizations of the linear layer. Vertical lines represent task transition. The red vertical line represents the end of the first training epoch.

Results: Comparison Gradient Based Methods with Similarity Based Methods

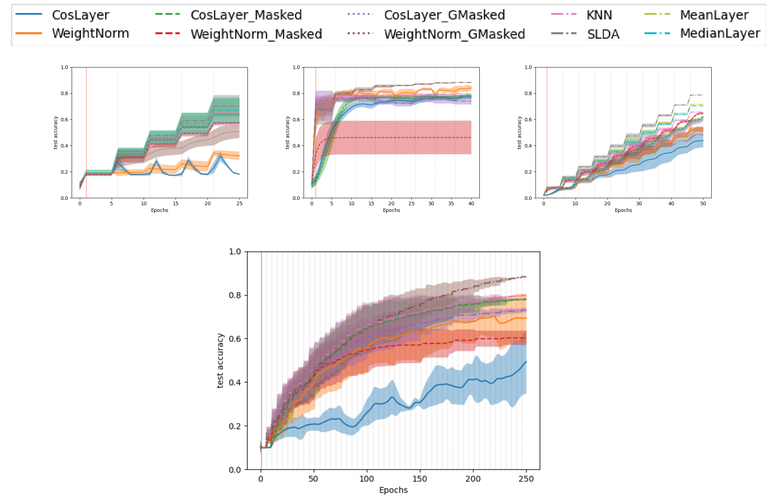

Fig. 4: Comparison between layers trained by gradient descent and layers trained without gradients. Experiments on CIFAR10, Core50, Core10Lifelong and Core10Mix on 8 different task orders (cf Appendix **??**). We plot the test accuracy on the full test set for each epochs.

The results of the study that the proximity-based methods are the best performing ones. Among the reparametrizations of the linear layer, the best performing methods were the proposed *weightnorm* and coslayer with the proposed *single masking* strategy (Incremental Setting). In the lifelong setting, alsmost all reparametrizations of the linear layer are well working, however, in this case the masking seems to bring instability in training.

Conclusion

In this paper, we conduct an empirical evaluation of the various output layers of deep neural networks. This evaluation is the first to be conducted where output layers are evaluated independently from feature extractor training. It gives us clear insights into how output layers learn continually. We also showed how different data distribution drift might affect the output layers differently in lifelong and incremental settings.
We describe three different factors that might cause catastrophic forgetting in a linear output layer in a continual classification task:

1. The modification of important weight (forgetting)
2. The interferences between output vectors
3. The projection drift: the feature extractor learn new features that change the representation space and may make the output vectors unsuitable.

We review different methods that aim to address the issues (1) and (2), and we also proposed a simplified version of weight norm and a masking strategy that improves the baselines.